How to build the MIIC OPeNDAP plugin

Follow these instructions to build the MIIC C++ plugin as a redhat RPM for distribution.

Required Environment

The current plugin build environment is a redhat 6.6 machine including:

1. gcc version 4.4.7 or better
2. GNU autoconf/automake tools
3. Boost libraries installed (sudo yum install boost & sudo yum install boost-devel)
4. Libdap & BES development libraries installed (see How to setup & run the MIIC OPeNDAP Server)

> For this project, the server "miic1" is currently functioning as a build machine. This may change.
>
> Note that you have to build as local user "miic". The LDAP account IDs are very large and appear to break the "tar" command which is used during building of the RPM.

Checkout MIIC Plugin Project

Use git to clone the repository:

```
% git clone https://git.earthdata.nasa.gov/scm/mii/multi-instrument-intercalibration.git
```

https://git.earthdata.nasa.gov/scm/mii/multi-instrument-intercalibration.git

Configure

The plugin builds from repository folder *multi-instrument-intercalibration/OPeNDAP/hyrax_plugins/miic*.

The build process uses autoconf and automake to generate the correct Makefile:

```
%  libtoolize
% automake --add-missing
% autoreconf
% ./configure
```

In new checkouts, the *automake* and *autoreconf* commands might report errors. There may be a strange chicken-egg problem with these tools but I find that just running the commands a 2nd time makes the errors go away.

> **configure failed... now what?**
> Oddly enough, configure will tell you *exactly* what went wrong in the file "config.log". You should open it and study it for the answer. For example, a build on a new machine reported this error in config.log:

```
configure:15853: $PKG_CONFIG --exists --print-errors "libdap >= $dap_min_version"
```

```
Package libdap was not found in the pkg-config search path.

Perhaps you should add the directory containing `libdap.pc'

to the PKG_CONFIG_PATH environment variable
```

Notice how it even told me how to fix the problem!
This particular problem was fixed by setting the env var PKG_CONFIG_PATH to /usr/lib64/pkgconfig. Although I really don't know why pkg-config didn't look there in the first place... it seems kind of standard.

Build the RPM

Finally, you can now build the RPM with the command "make RPM".

This should create a folder "rpmbuild" in your home directory to build the RPM.

The RPM will be located at rpmbuild/RPMS.

Odd tar behavior...
When building the RPM on a CENTOS 6 machine I was getting some very strange tar-related errors. It seems part of the RPM building process is to create tar files. Anyway the error was something like "uuid too large".

Anyway, it so happens that my user id on this system (id -u) was into **9 digits**. A uuid this large caused tar to explode. My workaround was to crate a new user "miic" with a much more reasonable 5 digit uuid and run the build as that user.

- How to setup MIIC development environment